

TITLE

**DISTRIBUTED ROUTER FOR DYNAMICALLY MANAGING
FORWARDING INFORMATION AND METHOD THEREOF**

CLAIM OF PRIORITY

[0001] This application claims priority under 35 U.S.C. §119 to our application entitled *DISTRIBUTED ROUTER FOR DYNAMICALLY MANAGING FORWARDING INFORMATION AND METHOD THEREOF* earlier filed in the Korean Intellectual Property Office on the 30th day of January 2003 and there assigned Serial No. 2003-6435, the contents of which are incorporated herein by reference.

BACKGROUND OF THE INVENTION

Field of the Invention

[0002] The present invention relates generally to a method of managing forwarding information in a distributed router (*i.e.*, in a router constructed with a distributed architecture), and more particularly, to a method of sharing in real time routing information collected at each router node from among all the router nodes within a distributed router and of dynamically managing forwarding information by aggregation based on the routing information.

Description of the Related Art

[0003] Along with the development of ultra high-speed, large-capacity networks, routers have

1 evolved from existing centralized architecture to new distributed architecture.

2 **[0004]** In a centralized router, a central processor implements routing protocols and manages
3 routing information collected by the routing protocols. For example, the central processor
4 computes routing tables and distributes to line cards the routing tables that it computes. Hence,
5 the line cards perform packet forwarding based on the routing tables computed by the central
6 processor.

7 **[0005]** On the other hand, a distributed router allocates the tasks of the central processor among
8 a plurality of processors, and consequently processes a large volume of data in comparison to a
9 centralized router. The distributed router has different processors; for example, a first processor
10 that manages the routing protocols, a second processor that calculates the routing tables, and a
11 third processor that manages packet forwarding. Each of the processors is dedicated to its assigned
12 function, thereby endeavoring to improve routing performance.

13 **[0006]** Routing nodes have their own routing tables to support their sub-networks and their own
14 processors to process routing paths. From the perspective of users, the routing nodes are seen as
15 one router, even though these routing nodes each globally manage the routing tables of the other
16 routing nodes within the distributed architecture of the router.

17 **[0007]** The input/output processor for each routing node is divided into a system processor area
18 with the routing protocols, a management processor, and a routing table, and a network processor
19 area with a forwarding table. The system processor for each of the routing nodes performs a set
20 of operations by collecting the routing information, managing the forwarding table obtained by
21 calculating a routing path, and sharing the routing table with the other input/output processors,

1 whereas the network processor forwards data between network equipment connected in the local
2 area in accordance with the forwarding table. Hence, the distributed router must process a large
3 volume of data rapidly.

4 **[0008]** Each of the routing nodes must share its forwarding table with the other routing nodes
5 in order to process a large volume of data rapidly in the distributed router. By way of example,
6 traditionally, each routing node exchanges the forwarding tables with the other routing nodes so
7 that any one of the routing nodes can globally manage the forwarding tables of all of the routing
8 nodes. Consequently, each routing node of a distributed architecture router must be equipped with
9 a large-capacity memory in order to store the forwarding tables; this creates an unnecessary
10 overhead for the main operational utility of routers, namely packet forwarding.

11 SUMMARY OF THE INVENTION

12 **[0009]** An object of the present invention is to substantially solve at least the above problems
13 and/or disadvantages and to provide at least the advantages below. Accordingly, an object of the
14 present invention is to provide a distributed router and process for preventing an increase in inter-
15 node traffic caused by packet forwarding attributed to each routing node's sharing of routing
16 information with the other routing nodes within a distributed architecture router.

17 **[0010]** Another object of the present invention is to provide a distributed architecture router and
18 a process in which each routing node shares in real time its collected routing information with the
19 other routing nodes within a distributed architecture router.

20 **[0011]** Still another object of the present invention is to provide a router and a process within

1 a distributed architecture router to manage forwarding information in a manner that reduces the
2 size of the forwarding table required at each routing node in a distributed architecture router.

3 **[0012]** Yet another object of the present invention is to provide a router and a process to
4 manage forwarding information for each routing node in a distributed router in which the routing
5 node shares in real time its collected routing information with the other routing nodes.

6 **[0013]** Still yet another object of the present invention is to provide a router and a process to
7 manage forwarding information by using a binary aggregation tree in each routing node, with the
8 aggregation tree being built with nodes corresponding to routing information and delegation nodes
9 that aggregate the routing information, in a distributed router.

10 **[0014]** A further object of the present invention is to provide a router and a process to
11 efficiently manage forwarding information by variably setting the aggregation level of delegation
12 nodes.

13 **[0015]** In accordance with an embodiment of the present invention, in a distributed architecture
14 router and process for managing forwarding information in the distributed router, all routing nodes
15 share a routing table that includes routing information collected by each of those routing nodes
16 and an aggregation tree created on the basis of the routing table. When new routing information
17 is inserted into the routing table, a position in the aggregation tree is detected at which an insertion
18 node corresponding to the new routing information is to be inserted into the aggregation tree. The
19 aggregation tree is searched to locate an ancestor node of the insertion node at, or below, a
20 predetermined maximum aggregation level. Given the presence of the ancestor node, the
21 forwarding table is left un-updated with information about the insertion node when both

forwarding information corresponding to the ancestor node is in the forwarding table with the ancestor node and the insertion node and the ancestor node were generated from the same source area. In the absence of the ancestor node, the aggregation level is reset to a level less than or equal to the maximum aggregation level, and a delegation node representative of the insertion node is inserted at the reset aggregation level. The source area of the inserted routing information is determined. If the source area of the routing information is a virtual area, forwarding information corresponding to the delegation node is inserted in the forwarding table. If the source area of the routing information is a local area, forwarding information corresponding to the insertion node is inserted into the forwarding table.

[0016] In accordance with another embodiment of the present invention, in a distributed router and a process to manage forwarding information in the distributed router in which all routing nodes share a routing table including routing information collected by each of the routing nodes and an aggregation tree created on the basis of the routing table. When routing information is deleted from the routing table, a deletion node corresponding to the deleted routing information is located in the aggregation tree. If forwarding information corresponding to the deletion node is in the forwarding table, the aggregation tree is searched to locate a descendant node of the deletion node at or below a predetermined maximum aggregation level. When a descendant node exists for the deletion node, the descendant node is set as a new source node of a delegation node. When no descendant nodes exist for the deletion node, forwarding information corresponding to the deletion node is deleted from the forwarding table.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] A more complete appreciation of the invention, and many of the attendant advantages thereof, will be readily apparent as the same becomes better understood by reference to the following detailed description when considered in conjunction with the accompanying drawings in which like reference symbols indicate the same or similar components, wherein:

[0018] FIG. 1 is a schematic block diagram that illustrates the configuration of an exemplary distributed router;

[0019] FIG. 2 is a schematic block diagram illustrating a typical process structure incorporated into a distributed router;

[0020] FIG. 3 is a schematic block diagram illustrating a process structure in a distributed router constructed as an embodiment of the present invention;

[0021] FIG. 4A illustrates the format of the management data about each node in an aggregation tree created to manage entries of forwarding information in an embodiment of the present invention;

[0022] FIG. 4B illustrates an aggregation tree created to manage entries of forwarding information in an embodiment of the present invention;

[0023] FIG. 4C is a two coordinate graph illustrating a prefix distribution as a function of the length of a prefix;

[0024] FIG. 5 is a flowchart illustrating a method for managing added routing information according to the principles of the present invention;

[0025] FIGs. 6A, 6B and 6C are flowcharts illustrating a method for adding entries of

forwarding information according to the principles of the present invention;

[0026] FIG. 7 is a flowchart illustrating a method for managing deleted routing information according to the principles of the present invention;

[0027] FIGs. 8A to 8G collectively illustrate an algorithm for dynamically managing entries of forwarding information according to the principles of the present invention;

[0028] FIGs. 9A and 9B are diagrams illustrating a method for managing added entries of forwarding information in an embodiment of the present invention;

[0029] FIGs. 10A and 10B are diagrams illustrating a method of managing deletion of forwarding information in an embodiment of the present invention; and

[0030] FIGs. 11A is a table and FIGs. 11B to 11D are two coordinate graphs that illustrate test results which verify the effects of managing dynamic forwarding information according to the principles of the present invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0031] Exemplars of current practice and preferred embodiments of the present invention will be described herein below with reference to the accompanying drawings. In the following description, well-known functions or constructions are not described in detail since they would obscure the invention in unnecessary detail.

[0032] FIG. 1 is a schematic block diagram of an exemplary router 100 having a distributed architecture. The distributed router illustrated in FIG. 1 may be a Galaxy Internet protocol (*i.e.*, an IP router).

[0033] Referring to FIG. 1, the distributed architecture of router 100 provides a plurality of routing nodes 110, 120, 130 and 140 connected to each other through a switching module (SWM) 150. Each of the routing nodes 110, 120, 130, 140 is provided with an input/output processor (IOP) 111. Each input/output processor 111 is designed to receive packets from two physical medium devices (PMDs: PMD-1 and PMD-2), respectively 112 and 113.

[0034] These routing nodes RNs 110, 120, 130, 140 each has its own routing tables to support their sub-networks and each has its own processors to process routing paths. Routing nodes RNs 110, 120, 130, 140 each run the routing protocols and perform forwarding, independently. From the perspective of users, routing nodes RNs 110-140 are seen as one router. Thus, routing nodes RNs 110, 120, 130, 140 each manage the routing tables of the other routing nodes 110, 120, 130, 140, globally. A physical sub-network connected to any one of the routing nodes RNs 110, 120, 130, 140 is called a local area B, and a network formed by the connections between the routing nodes RNs 110, 120, 130, 140 via switching module SWM 150 is called a virtual area A.

[0035] FIG. 2 is a schematic block diagram illustrating a typical process structure for a distributed router 100, especially illustrating the process performed by input/output processors IOP 10, 20, 30, 40 in each routing node RN and the connections between switching module SWM 50 and IOPs 10, 20, 30, 40 (IOP#1, IOP#2, IOP#3 and IOP#4) in the case where distributed router 100 is equipped with four routing nodes RNs.

[0036] Continuing to refer to FIG. 2, IOP#1 includes a plurality of routing protocols, such as ripd 11, ospfd 12, bgpd 13 and isisd 14, an IOP management processor, Galaxy Loosely Unified Environment Daemon (*i.e.*, GLUED) 15 that serves as an input and output management processor,

a routing table 16, and a forwarding table 17.

[0037] Routing protocols 11, 12, 13, 14 collect routing information based on their unique collection criteria. Routing table 16 stores the collected routing information, and forwarding table 17 stores forwarding information obtained from calculating the stored routing information.

[0038] Management processor 15 of IOP#1 is responsible for storing the collected routing information in routing table 16 and the forwarding information in forwarding table 17, and managing routing table 16 and forwarding table 17. Management processor 15 also advertises the collected routing information stored by routing table 16 to the other input/output processors IOPs 20, 30 and 40 via switching module SWM 50.

[0039] IOP#1 is divided into a system processor area 60 with the routing protocols 11, 12, 13, 14, IOP management processor 15, and routing table 16, and a network processor area 70 with forwarding table 17. System processor area 60 performs a set of operations for collecting the routing information, managing the forwarding table assembled by calculating a routing path, and sharing the routing table with the other input/output processors IOPs 20, 30 and 40, whereas network processor area 70 performs forwarding work between network equipment connected in the local area according to the information stored in forwarding table 17. Thus, a distributed router, that is, a router constructed with a distributed architecture, can theoretically process a large volume of data more rapidly.

[0040] As illustrated in FIGs. 1 and 2, each routing node RN 10, 20, 30, 40 must share its forwarding table with the other routing nodes RNs 10, 20 30, 40 in order to process a large volume of data rapidly in distributed router 100. For this purpose, traditionally, each of the routing nodes

RN 10, 20, 30, 40 exchanges the forwarding tables with the other routing nodes RNs via switching module SWM 50 so that each of the routing nodes can globally manage the forwarding tables of all of routing nodes 10, 20, 30, 40. On the assumption that a distributed router 100 includes ten routing nodes and that each of the routing nodes has ten thousand forwarding entries, each of the routing nodes must manage one hundred thousand forwarding entries. This concomitantly necessitates a large-capacity memory:

$$\# \text{Memory capacity} = 10,000 \text{ forwarding entries per routing node} \bullet 10 \text{ routing nodes. (1)}$$

in order to store the forwarding tables containing in excess of one hundred thousand forwarding entries; we have found the need for memory capacity to create an unnecessarily large overhead for packet forwarding transmission.

[0041] FIG. 3 illustrates a process structure in a distributed router constructed as an embodiment of the present invention. In general, the distributed router typically has a plurality of input and output processors, referred to as IOPs such as IOP #1. The IOPs exchange information with one another via a single switch module, SWM 250. This architecture is simplified to illustrate only one IOP 210 (IOP#1) and a SWM 250 in the representation provided by FIG. 3, for conciseness, although other routing nodes 220 (IOP#2), 230 (IOP#3) and 240 (IOP#3) are contemplated.

[0042] Referring again to FIG. 3, system processor area 260 is a virtual private network (*i.e.*, a “VPN”). The process structure of the distributed router constructed as one of the several possible embodiments of the present invention may include an aggregation tree 218 in the system processor area 260 of IOP#1. Aggregation tree 218 is built with nodes corresponding to

1 forwarding information and routing information managed by each input and output processor IOP
2 in the distributed router and delegation nodes that aggregate the forwarding and routing
3 information, as disclosed in Korea Patent Application No. 2002-0075701 entitled *DYNAMIC*
4 *MANAGEMENT METHOD FOR FORWARDING INFORMATION IN ROUTER HAVING*
5 *DISTRIBUTED ARCHITECTURE*, and filed by the same applicant of this application and
6 subsequently filed in the United States Patent & Trademark Office on the 1st day of December
7 2003 and there assigned Serial No. 10/724,085, which application is incorporated herein.

8 **[0043]** U.S. Serial No. 10/724,085 describes a method for reducing the sizes of a forwarding
9 tables managed by each routing node of a router constructed with a distributed architecture; a
10 technique to reduce internal traffic by reducing transmission of control packets that are transmitted
11 in order to update a forwarding table in a router constructed with a distributed architecture; and
12 a process for dynamically managing forwarding information in response to an addition or a
13 deletion of routing information in a router having a distributed architecture, by aggregating or
14 disaggregating forwarding information.

15 **[0044]** As described by U.S. Serial No. 10/724,085 forwarding information may be managed
16 in a router constructed with a distributed architecture including a plurality of routing nodes, by
17 forming an aggregation tree corresponding to each routing node, with the aggregation tree
18 including nodes corresponding to forwarding information for each of the routing nodes and virtual
19 nodes for aggregating forwarding information for each of the routing nodes; varying the
20 aggregation tree when forwarding information is added to each of the routing nodes; identifying
21 the creation area of forwarding information added to each of the routing nodes; making an analysis

1 of the aggregation tree, advertising forwarding information to other routing nodes based on the
2 analysis, and storing forwarding information in a local forwarding table of a corresponding routing
3 node when the forwarding information is created in a local area of the corresponding routing node;
4 and making an analysis of the aggregation tree, and storing forwarding information in the local
5 forwarding table of the corresponding routing node based on the analysis when forwarding
6 information is not created in the local area of the corresponding routing node.

7 **[0045]** U.S. Serial No. 10/724,085 teaches that forwarding information may also be managed
8 in a router constructed with a distributed architecture including a plurality of routing nodes, by
9 forming an aggregation tree corresponding to each routing node, with the aggregation tree
10 including nodes corresponding to forwarding information for each of the routing nodes and virtual
11 nodes for aggregating forwarding information of each of the routing nodes; making an analysis
12 of the aggregation tree of each of the routing nodes in response to a deletion of forwarding
13 information in each routing node and identifying the creation area of deleted forwarding
14 information; advertising the deletion of forwarding information to other routing nodes only when
15 the deleted forwarding information is advertised to other routing nodes by analyzing the
16 aggregation tree when deleted forwarding information is created in a local area of the
17 corresponding routing node, deleting the node corresponding to deleted forwarding information
18 from the aggregation tree, and deleting forwarding information from a local forwarding table of
19 the corresponding routing node; and deleting the node corresponding to forwarding information
20 from the aggregation tree when deleted forwarding information is not created in the local area of
21 the corresponding routing node.

1 **[0046]** Switch module SWM 250 is provided with routing protocols ripd 251, ospfd 252, bgpd
2 253 and Dglued 254. Internal peer connections are made between these routing protocols 251 to
3 254 and their counterparts, ripd 211, ospfd 212, bgpd 213 and glued 214 in input/output processor
4 IOP #1.

5 **[0047]** Accordingly, switch module SWM 250 shares in real time routing information collected
6 by the respective routing protocols of each input/output processor IOP with IOP#1 and provides
7 the routing information to the IOPs of the other routing nodes via the internal peer connections.
8 Consequently, in a distributed router constructed according to the principles of the present
9 invention, the internal peer connections between the routing protocols make it possible to share
10 the same routing information among all input and output processors IOPs.

11 **[0048]** In the embodiment illustrated by FIG. 3, routing table manager modules RTMs 215 and
12 255, respectively included in IOP#1 and switch module SWM 250 globally manage routing
13 information obtained from the routing protocols 211, 212, 213, 214, and the routing protocols 251,
14 252, 253, 254, and manage the priority levels of the routing information.

15 **[0049]** FIG. 4A illustrates the format of the management data about each node in an aggregation
16 tree created to manage forwarding information according to an embodiment of the present
17 invention. Referring to FIG. 4A, the management data of each node includes a PREFIX that is an
18 address to which forwarding information is to be forwarded, a LENGTH that states the length of
19 the PREFIX, an Aggregation_Level that indicates the aggregation level of the node, an INDEX
20 indicating whether the node is a delegation node or is a source node to which the delegation node
21 belongs (*e.g.*, a node that has created the delegation node), a TYPE that indicates the type of the

1 forwarding information, a SOURCE IOP that indicates the routing node RN that generated the
2 forwarding information, a FT FLAG that indicates whether the forwarding information has been
3 stored in a local forwarding table, and general link information in the binary tree, for example,
4 PARENT is an indicator that identifies a parent node, L_SUBTREE is a data field indicating a left
5 subtree, and R_SUBTREE is a data field indicating a right subtree.

6 **[0050]** The aggregation level is initially set to a default level (*i.e.*, a maximum level) by an
7 operator. It is variable to at or below the default level, depending on the router implementation.
8 That is, the aggregation level can be reset dynamically according to the network condition so that
9 a delegation node for aggregating routing information can be created at or below the dynamic
10 level.

11 **[0051]** INDEX indicates, in the case of a delegation node, the source node that created the
12 delegation node. INDEX provides the information by which the source node can be searched for
13 from the delegation node. For example, if a delegation node is derived from the right child node
14 under the left child node of the delegation node, INDEX for the delegation node is “01” since
15 generally, a left child node is “0” and a right child node is “1”.

16 **[0052]** TYPE indicates the type of the routing protocol (*e.g.*, BGP, SDPF, RIP, etc.) by which
17 the forwarding information is generated in an input and output processor IOP 210. For a
18 delegation node, TYPE is set to indicate delegation such that the forwarding information for the
19 delegation node is distinguished from actual forwarding information. The delegation type is lower
20 in priority level than any other type (*i.e.*, BGF, SDPF, RIP, etc.). Since a delegation node provides
21 virtual forwarding information, actual forwarding information generated by the routing protocols

1 is of higher priority.

2 **[0053]** To set the type of forwarding information, a routing node determines the source area in
3 which the forwarding information was generated. If it was generated from a local area, the routing
4 node RN sets the type of the forwarding information according to the type of the processor that
5 created the forwarding information. On the other hand, if the forwarding information was
6 generated from a virtual area, the routing node RN sets the type of the forwarding information in
7 a manner that indicates that the forwarding information is virtual. To do so, the routing node RN
8 analyzes the PREFIX of the forwarding information. If the PREFIX is a private Internet Protocol
9 (*i.e.*, an IP) address, the routing node RN considers that the forwarding information is from the
10 virtual area. Otherwise, the routing node RN considers that the forwarding information is from the
11 local area.

12 **[0054]** A private IP address refers to an address that is available in a local network, identifying
13 a node within the local network. Therefore, a private IP address is unavailable from a source that
14 is outside the local network. A distributed router generally assigns a private IP address to each
15 routing node in order to identify the routing nodes. Therefore, in the practice of the present
16 invention, the source area of the forwarding information is identified by use of a private IP address
17 for each of the routing nodes.

18 **[0055]** FIG. 4B illustrates an example of an aggregation tree created to manage forwarding
19 information in the practice of one embodiment of the present invention. The aggregation tree is
20 a binary tree. The updates of the aggregation tree due to insertion and deletion of routing
21 information will be described in detail later, with reference to FIGs. 9A through 10B.

1 **[0056]** Turning now to FIG. 4C, routing entries in a real border gateway protocol (*i.e.*, a “BGP”)
2 core routing table, which is used to verify the performance of the present invention, are
3 concentrated at prefix length of between 15 to 24 units, as shown in FIG. 4C. The analysis of this
4 prefix length distribution reveals that the algorithm overhead involved in dynamic aggregation and
5 retrieval of forwarding information can be reduced by limiting the aggregation level of a
6 delegation node to the default level.

7 **[0057]** In accordance with the present invention, when the Galaxy IP router illustrated in FIG.
8 1 is extended to a multi-rack configuration, the routing table manager (*i.e.*, RTM) 255 of switching
9 module SWM 250 in each rack manages the routing tables of all input/output processors (*i.e.*,
10 IOPs) under its management; this enables forwarding information aggregation between IOPs to
11 be easily expanded to the level of forwarding information aggregation between racks.

12 **[0058]** FIG. 5 is a flowchart illustrating a method for managing added routing information in
13 an embodiment of the present invention.

14 **[0059]** Referring to FIG. 5, if during step 1100, new routing information is generated and added
15 to the routing table in a routing node RN in the distributed router configured as illustrated in FIG.
16 3, during step 1200 routing node RN detects the position of an insertion node corresponding to
17 the added routing information (hereinafter, the node corresponding to the added routing
18 information will be referred to as an insertion node). Specifically, a binary search of the
19 aggregation tree from a root node is repeated until a node is detected that has a prefix of the same
20 length as that of the added routing information. When the position of the insertion node is
21 detected, then during step 1300 routing node RN determines whether a delegation node exists in

1 the position of the insertion node. If step 1300 determines that no delegation node exists at the
2 position of the insertion node, that is, if step 1300 finds that the position is empty then during step
3 2000 the insertion node corresponding to the added routing information is inserted at the position.

4 **[0060]** If during step 1300, the presence of a delegation node is detected at the position of the
5 insertion node, the forwarding information corresponding to the delegation node is deleted in step
6 1600 to instead permit an insertion of the insertion node in step 2000.

7 **[0061]** For this purpose, during step 1400 routing node RN determines whether there is a
8 left/right subtree for the delegation node. If the delegation node does not have a left/right subtree,
9 in step 1700 the routing node RN deletes the forwarding information corresponding to the
10 delegation node from its forwarding table. If during step 1400 the delegation node is found to have
11 a left/right subtree, the nodes at the left/right subtree must be reinserted in step 1500 before the
12 forwarding information corresponding to the delegation node is deleted in step 1600, since the
13 delegation node is a representative of the nodes at the left/right subtree. After an insertion of the
14 nodes at the left/right subtree in step 1500, the whole insertion procedure of FIG. 5 is performed
15 again. This is typically termed a recursive function. That is, in the presence of the left/right
16 subtree, a reinsertion of a left/right subtree is performed in step 1500 and then in step 1600 the
17 forwarding information corresponding to the delegation node is deleted from the forwarding table.

18 **[0062]** As described above, when the delegation node exists at the position of the insertion
19 node, the forwarding information corresponding to the delegation node is replaced by the insertion
20 node in steps 1400 through 1700.

21 **[0063]** FIGs. 6A, 6B and 6C illustrate insertion step 2000 in more detail.

[0064] Referring to FIG. 6A, before routing node RN inserts the forwarding information corresponding to the insertion node, during step 2001 the routing node searches for an ancestor node of the insertion node at or below the default level of the insertion node in order to detect a position at which to insert a delegation node representative of the insertion node.

[0065] If step 2003 fails to detect an ancestor node, a default search level is set in step 2005, by Equation (2) as illustrated by the algorithm set forth in Fig. 8C:

$$\text{Search Level}(\text{searchlevel}) = (\text{defaultlevel} / 2) - 1 \dots\dots\dots (2)$$

In step 2007, routing node RN determines whether an ancestor node of the insertion node exists at or below the search level established in step 2007 in order to reduce the occurrence of errors when inserting a delegation node for the insertion node.

[0066] If step 2007 establishes that there is no ancestor node of the insertion node at or below the search level established in step 2005, then in step 2170, the delegation node representative of the insertion node for the new routing information being inserted into the routing table, is inserted at the default level. If an ancestor node of the insertion node is detected during step 2007 at or below the search level, in step 2009 a search is made for descendant nodes of the ancestor node within the default level, that is at or below the default for the search level established in step 2005 in order to determine the presence or absence of descendant nodes for which the delegation node is the representative.

[0067] In the absence of any descendant node at or below the default search level, the

delegation node for the insertion node is inserted at the default search level in step 2170. In the presence of a descendant node however, the aggregation level is reset as a dynamic level in step 2130 and in step 2150 the delegation node is inserted at the dynamic level of the aggregation level established by step 2130. The aggregation level is related to the tree level; by way of example, with reference to Fig. 4B, the aggregation level between node #9 and node #2 is two. Preferably, the dynamic level is set as shown in Fig. 8C, by Equation (3) as:

$$\begin{aligned} \text{dynamiclevel} = & \text{GetNodePrefixLength}(\text{insertion node}) \\ & - \text{GetDistPrefixLength}(\text{insertionnode}, \text{descendant node}) \dots\dots\dots (3) \end{aligned}$$

where:

GetNodePrefixLength is a function for computing the prefix length of the insertion node, and

GetDistPrefixLength is a function for computing a position when the prefix lengths of the insertion node and the descendant node differ.

For example, if the prefix length of the insertion node is 18 and the prefix length of the descendant node is different at 19, the dynamic level is one, (that is, $1 = 19 - 18$) according to Equation (3)). Thus, the delegation node is generated at a level higher than the insertion node by one level.

[0068] After the insertion of the delegation node in steps 2005 through 2170, in step 2190 the routing node RN determines the source area of the routing information that is being inserted in order to determine whether to store the added routing information in the forwarding table or not. If step 2190 determines that the routing information is from the virtual area, the forwarding

1 information corresponding to the delegation node for the routing information is inserted in the
2 forwarding table in step 2210. If step 2190 determines that the routing information is from the
3 local area however, the forwarding information corresponding to the insertion node for the routing
4 information is inserted in the forwarding table in step 2230.

5 **[0069]** FIG. 6B illustrates an operation performed after steps 2001, 2003 determine that there
6 is an ancestor node for the insertion node of the insertion node.

7 **[0070]** Referring to FIGs. 6A, 6B and 6C collectively, upon detection during step 2003 of an
8 ancestor node of the insertion node, a determination is made by the *SelectDelegationLevel*
9 algorithm illustrated by Fig. 6C in step 2300 to establish a dynamic level at which to insert the
10 delegation node. The *SelectDelegationLevel* algorithm sets the delegation level to -1 through steps
11 2350, 2360, 2370 to reset the dynamic level in order to omit a forwarding table update, if the
12 forwarding table has already been updated with the ancestor node of the insertion node and both
13 the insertion node and the ancestor node were generated from the same source IOP.

14 **[0071]** The dynamic level is compared with zero in step 2410. A dynamic level of zero implies
15 that a delegation node representative for the insertion node can not be inserted into the aggregation
16 tree, and the forwarding information corresponding to the insertion node is directly inserted in the
17 forwarding table in step 2510.

18 **[0072]** If the dynamic level is determined in step 2410 to be less than zero, the source area of
19 the routing information is determined again in step 2530 and the forwarding table is updated in
20 step 2550 with the forwarding information corresponding to the insertion node only if the source
21 area of the routing information is determined in step 2530 to be the local area. If the routing

information is determined by step 2530 to be from the virtual area, the routing node RN jumps to the next step, omitting the forwarding table update. In this case, the aggregation process exerts an effect that advantageously tends to reduce the memory overhead that would be otherwise incurred by the forwarding tables.

[0073] If the dynamic level is determined in step 2410 to be greater than zero, the routing node RN inserts the delegation node at the dynamic level in step 2430. The routing node RN then determines the source area of the routing information in step 2450. If the source area is found in step 2450 to be the local area in step 2470, the routing node RN inserts the forwarding information corresponding to the delegation node into the forwarding table. If the source area is found during the local area however, the routing node RN inserts the forwarding information corresponding to the insertion node in the forwarding table in step 2490.

[0074] FIG. 6C illustrates the algorithm for step 2300 in more detail. To select the dynamic level at which to insert the delegation node, the aggregation level is reset in dependence upon whether step 2310 detects the presence or absence of a descendant node of the insertion node. In step 2320, a search is made for a descendant node of the insertion node. If step 2320 determines that a descendant node of the insertion node was detected in step 2310, the dynamic level, that is, the aggregation level is reset in step 2330 by Equation (3) using the prefix length of the insertion node and the position where the insertion node and the descendant node differ in prefix length. On the other hand, in the absence of the detection of a descendant node in step 2320, the delegation level for the ancestor node of the insertion node is set in step 2340 at the aggregation level of the delegation node.

1 **[0075]** Meanwhile, if during step 2350 the ancestor node of the insertion node is determined
2 to exist in the aggregation tree if in step 2360, and if the insertion node and the ancestor node are
3 also determined to have been generated from the same source IOP, then in step 2370 the dynamic
4 level is set to -1 in order to omit the forwarding table updating step.

5 **[0076]** FIG. 7 is a flowchart illustrating a method for managing deleted routing information
6 according to an embodiment of the present invention.

7 **[0077]** Referring to FIG. 7, in step 5100, when routing information is deleted from the routing
8 table of an routing node RN, the routing table manager RNB locates a node corresponding to the
9 deleted routing information (hereinafter, referred to as a deletion node) in step 5200. This step is
10 similar to the search for the position of the insertion node performed when making an insertion
11 of routing information. That is, in step 5200 a binary search in the aggregation tree from a root
12 node is repeated until a node is located that has a prefix length equal to that of the deleted routing
13 information.

14 **[0078]** Upon detection of the deletion node, in step 5300 the routing node RN determines
15 whether, or not, the deletion node is the source node of a delegation node. If the deletion node is
16 the source node, the forwarding information corresponding to the delegation node is changed to
17 that of the deletion node in step 5400 in order to effect the deletion of the deletion node by
18 deleting the delegation node.

19 **[0079]** In step 5500, the routing node RN determines whether the forwarding information
20 corresponding to the deletion node is in the forwarding table. If it is not, the deletion procedure
21 ends. On the other hand, if the forwarding information corresponding to the deletion node is

present in the forwarding table, a descendant node for the deletion node is searched for in step 5600 at or below the default level. If a descendant node is detected during step 5600, the descendant node is set as a new source node of the delegation node in step 5800, which might otherwise lead to the loss of a delegation node for the other descendant nodes when the delegation node represents a plurality of descendant nodes and one of those descendant nodes which is the source node of the delegation node is deleted.

[0080] In the absence of the detection during steps 5600, 5700 a descendant node of the deletion node at or below the default level in step 5900, the routing node RN deletes the forwarding information corresponding to deletion node from the forwarding table.

[0081] FIGs. 8A to 8G illustrate an algorithm for dynamically managing forwarding information according to an embodiment of the present invention.

[0082] FIG. 8A illustrates a pseudo code representation of the steps 1100, 1200, 1300, 1400, 1500, 1600 and 1700 illustrated in FIG. 5. FIG. 8B illustrates a pseudo code representation of step 1200 illustrated in FIG. 5 and step 5200 illustrated in FIG. 7. FIG. 8C illustrates a pseudo code representation of steps 2001 through 2210 and 2230 illustrated in FIG. 6A. FIG. 8D illustrates a pseudo code representation of steps 2150 and 2170 illustrated in FIG. 6A. FIG. 8E illustrates a pseudo code representation of steps 2300, 2410, 2430, 2450, 2470, 2490, 2510, 2530 and 2550 illustrated in FIG. 6B. FIG. 8F illustrates a pseudo code representation of steps 2300 through 2370 illustrated in FIG. 6C. FIG. 8G illustrates a pseudo code representation of steps 5100 through 5900 illustrated in FIG. 7. Since those steps in FIGs. 8A to 8G have been described in detail above, their description is not repeated here. The algorithm presented in the pseudo code shown

1 by FIGs. 8A to 8G is a merely exemplary application. Thus, many modifications may be made to
2 the algorithm, in the practice of the principles of this invention.

3 **[0083]** FIGs. 9A to 10B are diagrams illustratively showing the forwarding information
4 managing techniques, and especially the method of managing added and deleted forwarding
5 information under a default level of two, in accordance with the principles of the present
6 invention. Specifically, FIGs. 9A and 9B illustratively show the management of added forwarding
7 information, and FIGs. 10A and 10B illustratively show the management of deleted forwarding
8 information.

9 **[0084]** In aggregation trees illustrated in FIGs. 9A to 10B, PREFIX provides an address to
10 which the forwarding information corresponding to a node is to be forwarded, by identifying a
11 destination node. SOURCE IOP indicates the source IOP of the routing node that created the
12 insertion node, and FE indicates whether the forwarding information has been stored in a
13 forwarding table. FE, if marked, indicates the presence of the forwarding information in the
14 forwarding table. INDEX indicates the source node of the node to be forwarded if the node to be
15 forwarded is a delegation node. LEVEL is the aggregation level of the node to be forwarded.

16 **[0085]** Referring to FIG. 9A, when new routing information with a forwarding Prefix of three
17 is produced in an IOP 210 (IOP#1) with forwarding information stored in the Forwarding Table
18 217, a controller of IOP#1 inserts the routing information (*i.e.*, node 3) in binary aggregation tree
19 218 of IOP#1. Since the aggregation level of the insertion node is two, a delegation node with a
20 prefix of zero is added two levels higher than the insertion node with a prefix of three. Since the
21 source node of the delegation node with a prefix of zero is the insertion node 3 (that is, the

insertion node with a Prefix of “3”), INDEX for the delegation node is “00”. Information about insertion node 3 is stored in a forwarding table because both the insertion node 3 and the delegation node with a prefix of zero have the same source, namely routing node 210 IOP #1.

[0086] Meanwhile, the other routing nodes share the routing information of IOP #1 (*e.g.*, that is, the other routing nodes IOP#2, IOP#3, IOP#4, share aggregation tree 218 of routing node IOP#1) in the nature of the distributed router of the present invention so that each node shares its routing information stored in its routing table with the other nodes. In the illustrated case of FIG. 9A, an IOP 220 (IOP#2) shares the routing information stored in routing table 216 of IOP#1. The same will be applied to cases described hereinbelow.

[0087] Referring again to FIG. 9A, although IOP#2 has the same aggregation tree as that of IOP#1, it has different forwarding table contents. That is, IOP#1 stores information about insertion node 3 in its forwarding table 217 in accordance with step 2230 because insertion node 3 is new routing information that was produced in the local area serviced by IOP#1, whereas IOP#2 stores information about delegation node 0 in accordance with step 2210 because delegation node 0 is new routing information was produced in the virtual area shared by IOP#1 of routing node 210 and IOP#2 of routing node 220.

[0088] FIG. 9B illustrates a case where routing information corresponding to with a prefix of seven is produced in the local area of IOP#2, with the delegation node 0 already stored in the forwarding table.

[0089] A controller of IOP#2 detects the position of insertion node 7 and searches in accordance with step 2007 for an ancestor node for insertion node 7 according to the default aggregation level

1 of “2” established by step 2005. The controller of IOP#2 then detects a node 1 as the ancestor
2 node of insertion node 7. Node 1 is an ancestor node through which the insertion node 3 of IOP#1
3 is detected. Therefore, a delegation node for node 7 is inserted at node 4 which serves as the
4 delegation node of insertion node 7 in accordance with step 2170, and not at ancestor node 1. As
5 described above, the aggregation level is variable and the aggregation level, when reset in step
6 2130, is called a dynamic level in the present invention.

7 **[0090]** As illustrated in FIG. 9B, both insertion node 7 and ancestor node 4 are added to the
8 aggregation tree 218 of IOP#1, and the forwarding table of IOP#1 is updated with the delegation
9 node 4 of the node 7 because the source area of node 4 and node 7 were created by the virtual area
10 shared by routing nodes 210, 220 (*i.e.*, IOP#1, IOP#2).

11 **[0091]** Meanwhile, the controller of IOP#2 updates its forwarding table with information about
12 insertion node 7 because node 7 is from the local area serviced by routing node 220, that is, by
13 IOP #2.

14 **[0092]** A description will be made now of deletion of routing information according to the
15 present invention with reference to FIGs. 10A and 10B.

16 **[0093]** Referring to FIG. 10A, a delegation node 0 is representative of nodes 3, 4, 5 and 6 that
17 were generated in the local area of IOP#1. If the source node of node 0 is node 3, information
18 about nodes 3, 4, 5 and 6 is stored in the forwarding table of routing node 210, IOP#1, while only
19 information about the node 0 is stored in the forwarding table of routing node 220, IOP#2.

20 **[0094]** FIG. 10B illustrates deletion of node 3, which is the source node of delegation node 0,
21 from the aggregation trees of routing trees 210, 220 illustrated in FIG. 10A.

1 **[0095]** In the case where the source node of a delegation node is deleted, the forwarding
2 information corresponding to the deleted source node, that is, the forwarding information
3 corresponding to the deletion node is deleted from the forwarding table, and the source node of
4 the delegation node is changed, without any change in a virtual area in the present invention.

5 **[0096]** As illustrated in FIG. 10B, if the node 3 is the source node of delegation node 0, and
6 node 3 is deleted, the controller of IOP#1 changes the source node of delegation node 0 to node
7 4. Meanwhile, the controller of routing node 220, IOP#2 does not need to perform any particular
8 operation on the forwarding table of routing node 220, because only information about delegation
9 node 0 is stored in the forwarding table of routing node 220.

10 **[0097]** FIGs. 11A to 11D illustrate test results that verify the effects of the dynamic forwarding
11 information management according to the embodiment of the present invention.

12 **[0098]** The test results were achieved from a test in which 53,000 routing entries were inserted
13 and a predetermined proportion of the routing entries were flapped under the conditions of a
14 Galaxy system constructed with one switching module SWM and two routing node IOPs and
15 routing table entries in a core border gateway protocol router (*i.e.*, a “BGP” router) available from
16 a site <http://www.telstra.net/ops/bgptable.html> or <http://bgp.potaro.net/>.

17 **[0099]** FIG. 11A illustrates the result of a test performed to determine whether the sequence of
18 inserted forwarding entries affects the effects of the present invention. Sample entries arranged
19 in an ascending order are inserted while the order of the sample entries is randomly changed ten
20 times under six defaults levels. Referring to FIG. 13A, the number of the forwarding entries is
21 varied between 60 to 110. Accordingly, it is noted from the above test result that the aggregation

effect of the present invention is rarely affected by the insertion order of the forwarding entries.

[0100] FIG. 11B is a graph illustrating overhead measurements under six default levels. From FIGs. 11A and 11B, it is concluded that the present invention is most effective at or below a default level of 3, considering router performance and overhead.

[0101] FIG. 11C is a graph illustrating the numbers of forwarding entries remaining in a forwarding table when between 0% to 70% of 53,000 core BGP routing entries are withdrawn. Referring to FIG. 11C, about 26% to 77% of the forwarding entries are reduced when the core BGP entries are simply inserted with a 0% deletion.

[0102] FIG. 11D is a graph illustrating the performance of an actual Internet environment measured by use of an Agilent Router Tester. The Agilent Router Tester generally measures the performance of a router under a dynamic routing environment in which a network topology or network reachability continuously changes. The figure illustrates a convergence time as a function of the number of flap entries. The convergence time refers to time required for the router to reflect routing information changes in a forwarding table. This “BGP4 flap convergence time” test measures a traffic redirect convergence time defined as time required to replace a primary router with a new router when the primary router is dropped due to changes in the network topology or network reachability. The test was performed using three routing nodes RNs. Two of the three routing nodes RNs advertise the reachability of the same network behind the routing nodes RNs while the other routing node RN generates data traffic in the advertised network. From the convergence time measurements illustrated in FIG. 11D, the convergence time is reduced when only information about the source node of a delegation node is changed to minimize the

1 amendment of the forwarding table in deleting the actual source node of the delegation node.

2 **[0103]** In accordance with the present invention as described above, each routing node RN
3 shares its collected routing information in real time with the other routing nodes RNs and manages
4 forwarding information dynamically based on the routing information in a distributed router,
5 thereby obviating the need for packet forwarding to share the routing information between the
6 routing nodes RNs. The routing information is selectively updated in forwarding tables. Thus, a
7 forwarding table for each routing node RN is efficiently managed. Furthermore, the size of the
8 forwarding table in each routing node RN can be reduced since the forwarding information of
9 each routing node RN is managed in the form of a binary aggregation tree and the aggregation
10 level of a delegation node that aggregates node information corresponding to routing information
11 in the aggregation tree is variably set.

12 **[0104]** While the invention has been shown and described with reference to certain preferred
13 embodiments thereof, it will be understood by those skilled in the art that various changes in form
14 and details may be made therein without departing from the spirit and scope of the invention as
15 defined by the appended claims.